

LOAD 'N' RUN

RIVISTA DI PROGRAMMI SU CASSETTA PER IL TUO COMPUTER

16 GIOCHI & UTILITY tutti per SPECTRUM

MARLOWE INDAGA

THE STRONG BOX

CALCIOMANIA

EROI ANTICHI

LITTLE AKIKO

SPRITES

MUSICA

LUNE

INTERVENTO AEREO

MERGE SCREEN

MORSE CODE

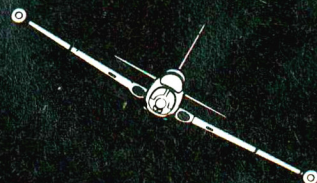
CARICATURE

STRIP DICE

O.S.T.A.R.

BASEBALL

PINA



B
U
O
N
A
N
N
O



ECCEZIONALE NOVITÀ

IN TUTTE LE EDICOLE

TUTTO QUEL CHE SERVE PER ENTRARE NEL FAVOLOSO MONDO DELLA COMPUTER-COMUNICAZIONE VIA TELEFONO

Hai un computer e un telefono? Questa rivista è fatta apposta per te! Tutto l'hardware e il software (su cassetta!) per comunicare via computer e telefono in Italia e in tutto il mondo. Potrai metterti in contatto con altri computer, trasmettere e ricevere posta, catturare programmi, trovare i mille amici (gli hackers!) del Modem Club International con cui scambiare programmi e informazioni... più una banca dati diretta a disposizione 24 ore... più altro ancora...

la prima rivista per computer via telefono

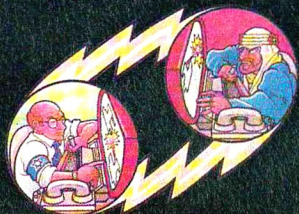
MODEM

COMPUTER MAGAZINE

N. 1 GENNAIO 1986

Sped. in abb. post. gr. III - L. 9.000

PIÙ
I NUMERI
DI TELEFONO
PER PORSI
IN
CONTATTO



CON IL
FASCICOLO,
GRATIS
IL SOFTWARE
SU
CASSETTA

COSTRUISCI IL TUO MODEM!

MODEM CLUB

DATA HACKERS NEWS

MICROMARKET LETTORI

LE INTERFACCE UTILI

MODEM MERCATO

da casa

**BANCA
DATI**

con il software
IN CASSETTA
OMAGGIO!



N. 23 - GENNAIO 1986

Direttore
Mario Magrone

Redattore Capo
Sira Rocchi

Direzione Tecnica
Nadia Marini

Stampa
Garzanti Editore S.p.A.
Cernusco S/N (MI)

Distribuzione
SO.DI.P. Angelo Patuzzi srl
Via Zuretti 25, Milano

Amministrazione, Redazione, Pubblicità: Arcadia s.r.l., C.so Vittorio Emanuele 15, 20122 Milano. Una copia lire 9.000, arretrato lire 10.000. Fotocomposizione: Composit. Selezione colori e fotolito: Eurofotolit. Stampa: Garzanti Editore S.p.A. Milano. Distribuzione: SO.DI.P. Angelo Patuzzi srl, Via Zuretti 25, Milano. Load 'N' Run è un periodico mensile registrato presso il Tribunale di Milano al numero 580 in data 24 dic. 83. Resp. Mario Magrone. Spedizione in abbonamento postale Gr. III/70. Pubblicità inferiore al 70%. Tutti i diritti sono riservati per tutti i paesi. Manoscritti, disegni, fotografie e programmi inviati non si restituiscono anche se non pubblicati. Rights reserved everywhere.

LOAD 'N' RUN

RIVISTA SU CASSETTA DI PROGRAMMI PER COMPUTER



SOMMARIO

- | | |
|---------------------|------------------|
| ■ PRONTO INTERVENTO | ■ EROI ANTICHI |
| ■ PINA | ■ LUNE |
| ■ LITTLE AKIKO | ■ STRIP DICE |
| ■ BASEBALL | ■ CARICATURE |
| ■ THE STRONG BOX | ■ MUSICA |
| ■ O.S.T.A.R. | ■ MARLOWE INDAGA |
| ■ MORSE CODE | ■ CALCIOMANIA |
| ■ SPRITES | ■ MERGE SCREEN |

Per caricare i programmi riavvolgete il nastro e date LOAD "", per uscire dai giochi togliete l'alimentazione e digitate nuovamente LOAD "". Le cassette utilizzate per registrare i programmi sono tutte di ottima qualità: assicuriamo comunque (rivolgetevi direttamente alla redazione *esclusivamente scrivendo*) la sostituzione di eventuali cassette difettose. Lo ZX Spectrum è un computer prodotto dalla Sinclair Research Ltd. (UK). È distribuito in Italia dalla GBC Italiana SpA.



LITTLE AKIKO (48K)

di V. Vernon

È l'antico gioco del sasso, della carta e delle forbici. La mano aperta indica la carta, che cede alle forbici ma che avvolge il sasso. La carta perciò perde contro le forbici ma vince contro il sasso. La mano con due dita indica la forbice che vince la carta ma perde contro il sasso. Infine il pugno indica il sasso, il quale perde contro la carta ma vince sulle forbici. Premendo «S» giocherai il sasso, con «F» le forbici e con «C» la carta. Per sette volte dovrai decidere cosa giocare e in base al gioco dell'avversario di volta in volta il computer deciderà il vincitore.

Il piccolo altoparlante dello Spectrum può emettere soltanto una nota per volta e l'istruzione che lo pilota è BEEP t,n, dove t è la durata della nota espressa in secondi ed n è il numero della nota da suonare.

Quando si deve inserire in un programma un brano musicale è necessario disporre di tutte le note, e relative durate, che compongono la melodia. La parte esecutiva della musica sarà quindi composta da più istruzioni BEEP o, come nel caso di questo programma, da un'unica istruzione BEEP interna a un ciclo FOR-NEXT che

```
100 RESTORE : FOR M=ZE TO UN: =
OR U=UN TO VAL "14": READ M$: BE
EP VAL ".125",VAL M$: PAUSE DU:
NEXT U: PAUSE DI: NEXT M: FOR S=
VAL "29" TO VAL "38": READ M$: B
EEP VAL ".125",VAL M$: PAUSE DU:
NEXT S
110 FOR I=VAL "39" TO VAL "44":
READ M$: BEEP VAL ".125",VAL M$
: PAUSE DU: NEXT I: RETURN
9999 STOP
1000 LET MAP=VAL "43900": LET MA
G=VAL "43939": LET GIO=VAL "4394
0": LET IP=VAL "48": LET POC=VAL
"213": LET CAR=VAL "56677": LET
FOR=VAL "56715": LET SAS=VAL "5
6736": LET POS=VAL "56102": LET
INC=VAL "56582": LET CIN=VAL "56
888": LET ZE=VAL "0": LET UN=VAL
"1": LET DU=VAL "2": LET TR=VAL
"3": LET QU=VAL "4": LET CI=VAL
"5": LET DI=VAL "10": LET SAL=V
AL "56544"
1030 DEF FN P$(R)=(("00" AND R=UN
)+("10" AND R=ZE)+("01" AND R=DU
): DEF FN F$(U$)=(("S" AND PAR), "
AND U$(UN)=U$(DU))+("A" AND YINTD
TU, " AND U$(UN)<U$(DU))+("AKIKO
" AND YINTD, " AND U$(UN)>U$(DU))+
"NO SIGNORE"
1040 POKE VAL "23609",VAL "25":
```

ha il compito di prelevare i dati ed emetterli sottoforma di suono. Vediamo appunto come è stato risolto il problema in questo programma.

Le linee 100 e 110 formano il sottoprogramma «musicale». Alla linea 100 l'istruzione RESTORE

inizializza il puntatore di lettura READ al primo blocco DATA presente nel programma (linea 1400). In questo DATA sono memorizzati i valori delle note che compongono il brano musicale. Successivamente nelle linee 100 e 110 più cicli FOR-NEXT nidifica-



PINA STORY (48K)

Aiuta l'indaffaratissima casa-alinga Pina a riordinare la sua casa. Per raccogliere gli oggetti sul pavimento devi prima prendere l'elettrodomestico dello stesso colore dell'oggetto posto sulla destra o sulla sinistra dello schermo. Per prendere un elettrodomestico devi avvicinare Pina ad esso. Muoviti velocemente altrimenti cadranno

per terra diversi oggetti che freneranno i tuoi movimenti.

Tasti:

A = su

Z = giù

N = sinistra

M = destra

P = pausa

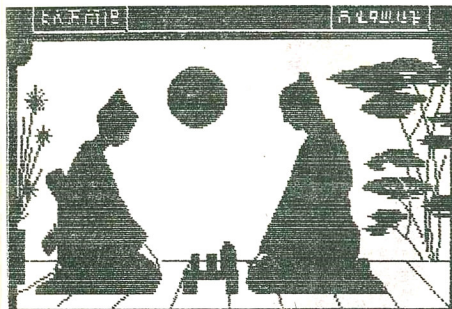
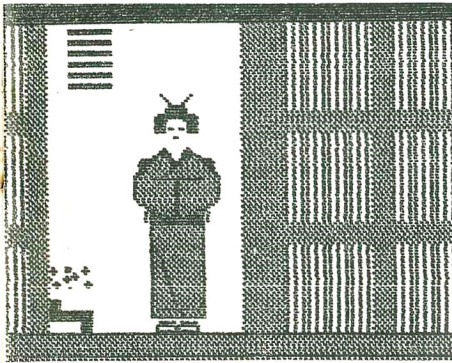
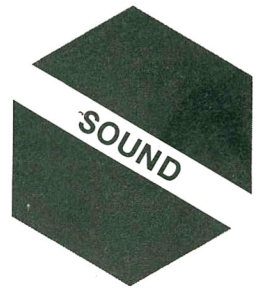
Q = fine gioco

S = pozione



MUSICA (48K)

di G. Colombo



ti leggono i dati delle note (istruzione READ MS) ed eseguono la melodia con l'istruzione BEEP VAL «.125», VAL MS. È da notare in quest'ultima istruzione che entrambi i parametri (durata e numero nota) sono espressi sotto forma di stringa.

Il grande maestro ZX si esibirà in due celeberrime sinfonie di J.S. Bach. Dopo l'apertura del sipario ti verrà presentato il menù. Premendo il tasto 1 potrai ascoltare la prima sinfonia intitolata «Jesu Decus»; premendo invece il tasto 2 ascolterai «Aria sulla 4° corda». Se vuoi sentire meglio la musica del tuo Spectrum collega solo la presa MIC al registratore e premi PLAY, ovviamente senza cassetta. Per il programma: le istruzioni che gestiscono una funzione di input o di output usano gli stream. Per esempio l'istruzione PRINT è equivalente a PRINT#2 perché invia i dati alla parte alta

dello schermo. Così LPRINT è equivalente a PRINT#3 e PRINT è uguale a LPRINT#2.

Risulta invece interessante il fatto di poter utilizzare PRINT#1 per scrivere nelle linee riservate all'input e all'edit. In questo modo è più facile accedere a questa parte dello schermo evitando di modificare alcune variabili di sistema. Andiamo ad osservare il listato e notiamo alla linea 110 l'istruzione PRINT#1; «seleziona la sinfonia desiderata».

Giustamente ha utilizzato il #1 per stampare nella parte bassa dello schermo.

```
1- RESTA CON NOI
2- ARIA sulla 4 cor
41 J.S.Bach
42 J.S.Bach
90 PRINT : PRINT : FOR n=1 TO
LEN A$: PRINT a$(n);: BEEP .005,
90: NEXT N
100 PRINT AT 19,0;"Per meglio a
scolltare collegami a un registr
store"
110 PRINT #1;"seleziona la sinf
onia desiderata"
120 IF INKEY$(">" "1" AND INKEY$("<"
"2" THEN GO TO 120
130 IF INKEY$="1" THEN CLS : GO
SUB 220
```



EROI ANTICHI (48K)

Nell'antica Grecia. Ti attendono tre duri cimenti: dovrai salire con balzi precisi le scoscese ed insidiose pendici dell'Olimpo evitando i massi che ti vengono scagliati contro. Dovrai quindi trovare la via d'uscita dal magico tempio del fatale Gordon e, infine, dovrai distruggere con le mazze in tuo possesso tutti i mostri che in-

contrerai ed aprire tutte le uscite del sotterraneo nel quale sei capitato. La serie di prove si ripeterà con l'aggiunta di altre difficoltà e nuovi letali protagonisti.

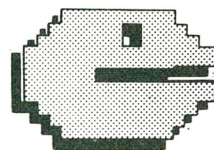
Tasti:

- O-P = destra/sinistra
- B = per saltare
- V = per abbassarsi



SPRITE (48K)

di M. Rigoni



L'utility permette di avere sullo Spectrum fino ad 8 sprites. Gli sprite sono formati da un carattere avente la dimensione di 2x2 «quadretti»; di questi caratteri ne abbiamo a disposizione 32, e possono essere richiamati per mezzo dell'apposito comando e stampati in una qualsiasi zona dello schermo. Possiamo anche raggruppare questi caratteri in una sequenza che verrà poi stampata sullo schermo; questa sequenza potrà essere lunga al massimo 4 caratteri e sarà a sua volta identificata da un numero che è appunto quello dello sprite.

Le nuove istruzioni devono, per essere accettate, essere precedute dal punto esclamativo (!) e seguite dai vari parametri numerici; inoltre i comandi !DOKE e !DATA accettano numeri esadecimali purché preceduti dal simbolo del dollaro.

Se il comando non può essere eseguito per qualche ragione, il programma si fermerà e riporterà il tipo di errore. Alcuni dei nuovi comandi hanno nuovi messaggi d'errore; per esempio un !UNTIL che non è stato preceduto da !REPEAT darà come errore «Mismatched until»; i nuovi comandi possono essere scritti sia in maiuscolo che in minuscolo.

Ecco le nuove istruzioni:

!GRAPHIC numero carattere:

Numero carattere deve essere un numero compreso tra 1 e 32. Questo comando provvede a salvare i dati del carattere contenuti nel comando !DATA.

!DATA linea1, linea2...:

Questo comando può essere seguito fino ad un massimo di 16 espressioni numeriche, una per ogni linea del carattere che è stato definito; ogni espressione numerica può essere un numero decimale (max 65535), un numero esadecimale, o un numero binario (BIN 1101).

!CHAR numero sprite, carattere1 (,carattere2...):

Questo comando, si usa quando si stampa lo sprite. Se si vuole lo sprite animato devi dare la sequenza dei caratteri (max 4), e una routine provvederà automaticamente all'animazione.

```
110 POKE 220550,0
120 LET Z#="R"
140 LET mdv=(z#="M")
150 IF mdv THEN PRINT "Metti i
cartridge nel drive 1"
160 GO SUB 9900
170 LET a#="))))))))))))))"
180 LET st=64280
190 PAUSE 100
```

L'intervallo tra un carattere e l'altro si regola con il comando !FRAMES.

!FRAMES numero sprite, ritardo:

Ritardo può assumere valori compresi tra 1 e 255, e controlla la velocità di animazione dello sprite in questione.

!ON EDGE numero sprite, azione sul bordo:

Questo comando controlla quando

```
4010 GO SUB 9100
4020 IF mdv THEN PRINT "Metti il
cartridge nel drive 1"
4030 INPUT "Nome del file"; LINE
f$
4040 IF f$="" OR LEN f$>10 THEN
GO TO 4030
4050 PRINT "SAVE ";f$
4060 PRINT "CODE 64280,1088"
4080 SAVE f$CODE 64280,1088
```

uno sprite raggiunge il margine dello schermo. Se «azione sul bordo» è uguale a 0, lo sprite continua il movimento rientrando dal lato opposto; se è uguale a 1 lo sprite si ferma; se è uguale a 2 lo sprite rimbalza tornando indietro.

!ON HIT Numero sprite, tipo collisione:

Se «tipo collisione» è uguale a 0, lo sprite continua il movimento sopra ogni oggetto sullo schermo; se è uguale a 1 lo sprite si ferma.

!SPRITE numero sprite,x,y:

Lo sprite è stampato sullo schermo; x e y sono le coordinate in pixel dall'angolo alto-sinistra dello sprite.

!SPRITE numero sprite,x,y,xmov., ymov., passo:

Lo sprite è stampato alla posizione (x,y) e si muove per un certo numero di passi (1-254). Esso si muove di xmov. e ymov. per ogni passo. (Valori possibili da -8 a +8. Se il passo è

uguale a 255 lo sprite si muoverà di continuo).

!ERASE numerosprite:

Lo specifico sprite è cancellato.

!ERASE O:

Cancella tutti gli sprite attivi.

!DI e !EI:

La routine di interrupt che muove gli sprite è fermata; !EI effettua il contrario.

!PUT numero carattere,x,y:

Lo specifico carattere grafico è stampato sullo schermo alla posizione (x,y).

!REPEAT; !UNTIL condizione:

Questi due comandi si trovano nel Pascal o nel Basic strutturato. Ogni comando compreso tra REPEAT e UNTIL sarà ripetuto finché la condizione non sarà verificata.

!DOKE indirizzo, valore:

Questo comando esegue un doppio POKE di un numero a 16 bit in due indirizzi consecutivi. Possono essere usati numeri esadecimali sia per l'indirizzo che per il valore.

!SET INK vecchio valore, nuovo valore:

Questo comando cambia gli inchiostri sullo schermo.

!SET PAPER vecchio valore, nuovo valore:

Come comando precedente.

!SCREEN colore carta, colore inchiostro:

Questo comando cambia il colore sullo schermo senza cambiarne il contenuto.

!ZAP:

Questo comando produce un suono tipo laser.

!NOISE lunghezza:

Questo comando produce un suono la cui durata dipende da lunghezza (1-255).

I 32 caratteri grafici sono salvati in memoria a partire dall'indirizzo 64280 per una lunghezza di 1088 byte. Si può quindi salvare l'intero set di caratteri più il programma in L/M con SAVE «nome file» CODE 60000,5367. Da notare che si hanno ancora a disposizione i 20 caratteri definibili dall'utente.

Per trovare la posizione dello sprite in ogni momento, o per esaminare se una collisione c'è stata, possono essere usate queste funzioni pre-definite, dove S è uguale al numero dello sprite (1 ÷ 8).

DEF FN x(S) = PEEK(63727+32*S)
: rem coordinata x

DEF FN y(S) = PEEK(63728+32*S)
: rem coordinata y

DEF FN c(S) = PEEK(63722+32*S)
: rem flag di collisione

Quest'ultima funzione può assumere 3 diversi valori: 0 se non c'è stata collisione; 1 se si è verificata una collisione con il bordo dello schermo; 128 se c'è stata una collisione con un altro sprite o con lo sfondo.

Abbinato al programma degli sprite interamente in L/M vi è un programma scritto in Basic per fare gli sprite.

Questo programma permette di fare, vedere, correggere, salvare, caricare, cancellare e animare gli sprite. Il programma è autoesplicativo.

Il programma può essere modificato per microdrive.

Alla linea 130 vi è l'assegnazione LET ZS=«R», dove R sta ad indicare il registratore. Per chi possiede il microdrive dovrà scrivere alla 130 LET ZS=«M». A questo punto la successiva istruzione (linea 140) porrà la variabile mdv a 1 ad indicare che il supporto di memoria di massa è il microdrive. Per il save la linea 4020 stampa il messaggio «Metti il cartridge nel drive 1» se la variabile mdv è a 1. Dopodiché viene chiesto il nome del file da salvare (linea 4030) e successivamente (linea 4080) viene salvato il file su cassetta. Per poter adattare la subroutine anche al microdrive è sufficiente inserire la seguente linea:
4070 IF mdv THEN SAVE «m»;1;
f\$CODE 64280,1088: GOTO 4090

Per load aggiungere la linea
4630 LOAD «m»;1;f\$CODE 64280,1088



BASEBALL

(48K)

di M. Vaccaroli

Ecco uno splendido gioco a squadre: la squadra ospite (colore blu) e la squadra dello Spectrum (colore rosso). Il gioco si divide in nove tempi (INNING). Ogni Inning è diviso in due parti. Nella prima parte la squadra ospite gioca in attacco. Il giocatore deve colpire la palla lanciata dall'avversario. Per fare questo bisogna premere ENTER quando la palla è vicinissima ai lui.

Si possono verificare tre possibilità:

Dopo quattro ball il battitore guadagna una base.

Se la palla è battuta, il battitore dovrà correre per raggiungere la base alternando i tasti 1 e 0; guadagnerà la base se arriverà prima che il difensore abbia ricevuto la palla. Quando le quattro basi sono tutte occupate, la squadra segna un punto.

Nella seconda parte dell'Inning la squadra ospite lancerà la palla e dovrà cercare d'impedire alla squadra dello Spectrum di occu-



by Marco V.

— Il battitore gira la mazza a vuoto oppure non colpisce la palla lanciata; in questo caso l'arbitro dirà STRIKE (penalità per il battitore).

— La palla lanciata dall'avversario è troppo alta o troppo bassa

pare la basi e segnare così dei punti.

Per lanciare si preme ENTER quando il lanciatore ha finito il caricamento.

Anche qui tre possibilità:
— sarà ball se il lanciatore pre-

```
9000>LET q=(32-LEN f$)/2
9010 FOR w=LEN f$ TO 1 STEP -1:
PRINT PAPER 4;AT 7,q) /$(w TO ):
BEEP :01,w: NEXT w
9015 IF f$="* * * BATTUTA * * *"
THEN POKE 56003,0: POKE 56006,9
: RANDOMIZE USA 56000: GO TO 903
0
9020 FOR w=1 TO 100: NEXT w
9030 PRINT PAPER 4;AT 7,1;"
": RETU
RN
```

per essere battuta; in questo caso l'arbitro dirà BALL (penalità per il lanciatore).

— La palla è battuta dal battitore.

Dopo tre strike il battitore è eliminato e, dopo tre eliminazioni, si cambia campo e la squadra ospite passa in difesa.

merà ENTER prima della fine del caricamento;

— sarà strike se invece premerà ENTER immediatamente dopo il caricamento;

— la palla sarà invece battuta dal battitore dello Spectrum se il lanciatore tarderà troppo nel premere

ENTER.

Se la palla è battuta, l'ospite diverrà il difensore e dovrà recuperare la palla correndo verso di lei il più velocemente possibile alternando i tasti 1 e 0. Eliminati tre battitori avversari, inizierà un nuovo Inning.

Vinca il migliore!

In questo programma vengono stampati più volte vari messaggi. Vediamo come l'autore ha impostato la loro visualizzazione.

La subroutine che esegue la stampa è compresa dalla linea 9000 alla linea 9030. Il messaggio da stampare è contenuto nella va-

riabile alfanumerica f\$. Una volta definita questa variabile si chiama la subroutine di stampa con l'istruzione GOSUB 9000. Alla linea 9000 viene calcolata la colonna dalla quale iniziare a stampare il messaggio. Il calcolo è memorizzato nella variabile q. Successivamente in un ciclo FOR-NEXT con step decrescente viene eseguita l'istruzione PRINT AT 7,q; f\$ (w to) la quale stampa alla riga 7 e colonna q la parte del messaggio f\$ che inizia dal carattere di posizione w fino alla fine. Segue l'istruzione BEEP che produce il suono di accompagnamento.



PAOLO
ROSSI



MICHEL
PLATINI



BRUNO
CONTI



PRONTO INTERVENTO (48K)

Una nuova minaccia insidia New York: terribili ufo si insinuano per le vie della città volando fra i grattacieli. Serve, per sgominarli, un eccezionale pilota di elicottero che sappia volare fra i grattacieli senza finirci contro e che sia anche un abile combattente.

Nella città ci sono delle piattaforme sotto il livello della strada sulle quali puoi atterrare per fare

pilota esperto saprai a cosa serve quel codice!

I tasti sono tutti ridefinibili.

Alla domanda Riga/O no? premere R oppure O.

Se scegli i tasti Riga essi sono:

1-0 = fuoco

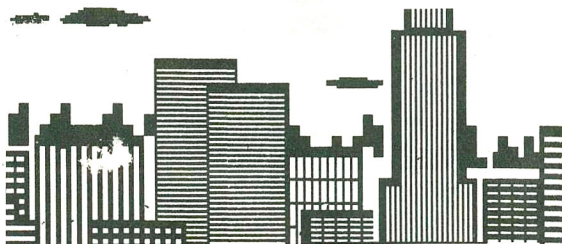
Y-P = alto

h-ENTER = basso

Q-T = più veloce

A-G = più adagio

Z-V = sinistra



rifornimento.

Può darsi però che la corte marziale ti giochi delle brutte sorprese...

All'inizio ti si chiederà un codice di livello. È sufficiente che tu prema «enter». Se diventerai un

B-SIMBOL SHIFT = destra

Due consigli: la luce gialla sotto il radar diventa rosa quando l'ufò può essere colpito dai tuoi razzi. Inoltre, stai attento alla nebbia: se capita di non vedere più niente scendi di quota.



STRIP DICE (48K)

di G. Stefani e M. Salis



MUNDIAL CARICATURE

(48K)

di C. & D. Cianfanelli



Ammira i tuoi amati eroi del calcio ridisegnati dalla mano del caricaturista. Conti, Platini, Maradona ed altri tutti insieme nel tuo Spectrum!

La parte principale del programma che gestisce la visualizzazione delle otto caricature risiede dalla linea 1005 alla linea 1010. In queste linee si eseguono le diverse chiamate alle subroutine compresa quella dedicata alla musica (GOSUB 2005). Quest'ultima parte dalla linea 2005 fino alla linea 2014. È costituita da un blocco DATA nel quale sono memorizzati i dati della durata e numero nota dell'istruzione BEEP. Il motivo musicale è composto da 30 note (ciclo FOR-NEXT) che vengono lette (READ c,d) ed eseguite (BEEP c,d) alla linea 2005. Sempre alla stessa vi è l'istruzione IF INKEYS<>" " THEN RETURN. Questa legge lo stato della tastiera (INKEYS) e se (IF) un tasto è stato premuto smette l'esecuzione della musica e ritorna (RETURN) al ciclo principale del

```

2005>RESTORE 2007: FOR f=1 TO 30
: READ c,d: BEEP c,d: IF INKEYS<>
: " " THEN RETURN
2006 NEXT f
2007 DATA 0.25,0.25,0.5,0.5,0.25,0.
5,0.25,0.25,0.5,0.5,0.25,0.25,0.
105,0.105,0.105,0.5,0.5,0.25,0.5,0.
0.25,12,0.75,14,0.5,14
2009 DATA 0.5,12,0.25,0.5,0.
0.5,0.5,0.5,0.5,0.5,0.5,0.25,
0.105,0.75,0.5,0.5,0.25,0.5,0.25,
0.75,5
2014 GO TO 2005
2020 PRINT AT 4,1; BRIGHT 1;P$;A
T 5,1;P$;AT 6,1;P$;AT 7,1;P$;AT
8,1;P$;AT 10,1;P$;AT 11,1;P$;AT
13,1;P$;AT 14,1;Q$;AT 15,1;Q$;AT
16,1;Q$;AT 17,1;Q$;AT 18,1;Q$;A
T 19,1;Q$
2022 BORDER 7: PRINT AT 20,1; BR
IGHT 1;Q$;AT 21,1;Q$: PRINT #1;
BRIGHT 1; INK 3;AT 0,1;"by Cianf
anelli " : RETURN
  
```

programma (linea 1006). Se invece nessun tasto è stato premuto allora la musica prosegue fino alla fine per poi riprendere dall'inizio (linea 2014 istruzione GOTO 2005).

Se volete far eseguire a tutti i costi per intero la melodia tra una

caricatura ed un'altra, modificate l'istruzione 2005 e 2006 come segue:

```

2005 RESTORE 2007: FOR f=1
TO 30: READ c,d: BEEP c,d:
2006 NEXT f: RETURN
ed ascoltate tranquillamente la
musichetta!
  
```

La visione di questo programma è sconsigliata ai minori ed ai malati di cuore! Perché?! Perché si tratta di una versione con i dadi del nostro strip poker.

All'inizio del gioco si fisserà il valore del piatto, che rimarrà tale per tutta la durata della partita. Quindi seguirà la prima puntata del giocatore, al quale verrà mostrato il primo dei due dadi a sua disposizione. Toccherà poi allo Spectrum scegliere se rilanciare, vedere o passare. Eventuali correzioni si possono fare premendo il solo tasto «0». Vediamo ora qualcosa degli UDG.

Lo Spectrum offre differenti vantaggi sotto il profilo grafico.

UDG A = █	UDG H = █	UDG O = █
UDG B = █	UDG I = █	UDG P = █
UDG C = █	UDG J = *	UDG Q = █
UDG D = █	UDG K = *	UDG R = █
UDG E = █	UDG L = █	UDG S = █
UDG F = █	UDG M = █	UDG T = █
UDG G = █	UDG N = █	UDG U = █

Uno fra questi è la possibilità di definire 21 caratteri detti UDG (user defined graphics). I codici di questi nuovi caratteri sono da 144 a 164.

In questo programma i 21 UDG sono serviti a creare diverse picco-

le immagini, tra queste anche quella del piatto.

Normalmente i dati degli UDG si collocano a partire dall'indirizzo USR «a», che per il 48K corrisponde al valore 65368.

Vi è una variabile di sistema,

chiamata appunto UDG (23675-23676), che indica l'inizio dove sono memorizzati i dati UDG. Accendendo uno Spectrum 48K e digitando PRINT PEEK 23675+256*PEEK 23676 verrà stampato il valore 65368 che sta ad indicare lo stesso di USR «a». Sappiamo però che il sistema permette, senza alcuna modifica, di definire solo 21 UDG: dalla lettera «A» alla lettera «U». Molto spesso si devono creare diversi disegni realizzati con caratteri definiti e 21 UDG non sono sufficienti. Per ovviare al problema è possibile modificare la

variabile di sistema UDG in modo tale da riservare spazio ai nuovi dati ed ottenere così più caratteri definiti. Supponiamo quindi di dover definire 42 caratteri. I dati dei primi 21 saranno memorizzati normalmente, a partire dall'indirizzo 65368. Gli altri 21 caratteri li memorizzeremo allo stesso modo dei primi avendo però preventivamente modificato la variabile UDG. Per far questo dobbiamo calcolare la quantità di spazio necessaria per memorizzare i nuovi 21 caratteri. Questa è pari a 21x8 bytes, cioè 168 bytes. A questo

punto dobbiamo calcolare il nuovo indirizzo da memorizzare in UDG. Esso è 65368-168=65200. Quindi eseguiremo le seguenti istruzioni: POKE 23675,65200-INT (65200/256)*256

POKE 23676,INT (65200/256)
 Con queste due istruzioni abbiamo abilitato il secondo banco dei 21 UDG. Per ripristinare il primo è sufficiente eseguire queste altre due istruzioni:
 POKE 23675,65368-INT (65368/256)*256
 POKE 23676,INT (65368/256)



O.S.T.A.R.

di R. Ciardo (48K)

In questo gioco tenterai la traversata atlantica in barca a vela da Plymouth a Newport. Sulla carta nautica la posizione di partenza sarà indicata da una crocetta, la destinazione da un quadratino lampeggiante. Dopo i preliminari della partenza, con delle frecce lampeggianti sarà visualizzata, per ogni giorno, la direzione del vento. In base ad essa deciderai in che direzione procedere.

Avviata la barca, il tuo alter ego a bordo ti informerà circa gli eventuali problemi della giornata, risolti i quali aggiornerà la posizione dell'imbarcazione secondo il cammino percorso. Se non sei proprio un asso della nautica, il programma è in grado di darti una mano. Un primo aiuto è possibile ottenerlo quando il computer chiede andature e mure; rispondendo «aiuto» sarà visualizzata una tabellina con delle freccette indicanti, per ogni andatura, la direzione

di avanzamento dell'imbarcazione. Il secondo «aiuto» è di tipo linguistico, nel senso che potrai chiedere il significato di parole sconosciute (per inciso, alla domanda «qual è la parola?» potrai anche rispondere «tutte» e il computer fornirà il suo vocabolario). Buon vento, amico! Avete intanto visto nel programma gli array?!

Uno dei concetti basilari della programmazione è quello di «array». Con il termine «array» s'intendono i vettori e le matrici. L'istruzione Basic che definisce un array è DIM. Nel Basic dello Spectrum questa istruzione si rivela assai potente e a seconda di come viene usata assume significati diversi.

Innanzitutto dobbiamo dire che con l'istruzione DIM possiamo definire sia array numerici che alfanumerici. Per esempio l'istruzione DIM c(10) definisce il vettore numerico c a 10 posizioni. Invece l'istruzione DIM c(10,10) dichiara la matrice numerica c composta da 10 righe e 10 colonne, per un totale di 100 posizioni. Differente è invece la dichiarazione di array alfanumerici. Per esempio DIM c\$(10) indica che è stato definito il vettore alfanumerico c\$ di 10 posizioni. In ogni posizione però è

possibile memorizzare un solo carattere. Provate infatti ad eseguire le seguenti istruzioni:

```
10 DIM C$(10)
20 LET C$(1)="LNR"
30 PRINT C$(1)
```

dopo aver indicato il RUN verrà stampato solo il carattere «L» ad indicare che in C\$(1) è stato memorizzato solo un carattere.

Utilizzare l'istruzione DIM con più parametri per array alfanumerici vuol dire interpretare l'istruzione in due modi differenti. Supponiamo per esempio di eseguire il comando DIM C\$(10,10), in questo caso abbiamo indicato che il vettore alfanumerico c\$ è di 10 posizioni ognuna avente la possibilità di memorizzare 10 caratteri; oppure è possibile interpretare DIM c\$(10,10) come la matrice c\$ di 10 righe per 10 colonne nella quale ogni locazione può contenere un solo carattere.

Vediamo come l'autore di questo programma ha utilizzato questa istruzione. Alla linea 10 viene dimensionato il vettore v\$(8,2) e successivamente vengono assegnate alle 8 differenti locazioni del vettore i diversi valori alfanumerici. In questo caso quindi l'array v\$(8,2) è utilizzato come vettore.

```
2 RANDOMIZE
10 DIM V$(8,2): LET V$(1)="N":
LET V$(2)="$": LET V$(3)="E":
LET V$(4)="O": LET V$(5)="NE":
LET V$(6)="NO": LET V$(7)="SE":
LET V$(8)="SO"
20 DIM V$(10,2): LET V$(1)="N":
LET V$(2)="$": LET V$(3)="E":
LET V$(4)="O": LET V$(5)="NE":
LET V$(6)="NO": LET V$(7)="SE":
LET V$(8)="SO": LET V$(9)="NO":
LET V$(10)="SE": LET V$(11)="SE":
LET V$(12)="SE": LET V$(13)="SE":
LET V$(14)="SE": LET V$(15)="SE":
LET V$(16)="SE": LET V$(17)="SE":
LET V$(18)="SE": LET V$(19)="SE":
LET V$(20)="SE": LET V$(21)="SE":
LET V$(22)="SE": LET V$(23)="SE":
LET V$(24)="SE": LET V$(25)="SE":
LET V$(26)="SE": LET V$(27)="SE":
LET V$(28)="SE": LET V$(29)="SE":
LET V$(30)="SE": LET V$(31)="SE":
LET V$(32)="SE": LET V$(33)="SE":
LET V$(34)="SE": LET V$(35)="SE":
LET V$(36)="SE": LET V$(37)="SE":
LET V$(38)="SE": LET V$(39)="SE":
LET V$(40)="SE": LET V$(41)="SE":
LET V$(42)="SE": LET V$(43)="SE":
LET V$(44)="SE": LET V$(45)="SE":
LET V$(46)="SE": LET V$(47)="SE":
LET V$(48)="SE": LET V$(49)="SE":
LET V$(50)="SE": LET V$(51)="SE":
LET V$(52)="SE": LET V$(53)="SE":
LET V$(54)="SE": LET V$(55)="SE":
LET V$(56)="SE": LET V$(57)="SE":
LET V$(58)="SE": LET V$(59)="SE":
LET V$(60)="SE": LET V$(61)="SE":
LET V$(62)="SE": LET V$(63)="SE":
LET V$(64)="SE": LET V$(65)="SE":
LET V$(66)="SE": LET V$(67)="SE":
LET V$(68)="SE": LET V$(69)="SE":
LET V$(70)="SE": LET V$(71)="SE":
LET V$(72)="SE": LET V$(73)="SE":
LET V$(74)="SE": LET V$(75)="SE":
LET V$(76)="SE": LET V$(77)="SE":
LET V$(78)="SE": LET V$(79)="SE":
LET V$(80)="SE": LET V$(81)="SE":
LET V$(82)="SE": LET V$(83)="SE":
LET V$(84)="SE": LET V$(85)="SE":
LET V$(86)="SE": LET V$(87)="SE":
LET V$(88)="SE": LET V$(89)="SE":
LET V$(90)="SE": LET V$(91)="SE":
LET V$(92)="SE": LET V$(93)="SE":
LET V$(94)="SE": LET V$(95)="SE":
LET V$(96)="SE": LET V$(97)="SE":
LET V$(98)="SE": LET V$(99)="SE":
LET V$(100)="SE":
```




LOAD BANK

Hai programmi originali, esclusivamente pensati e fatti da te? Mandaceli in visione e, se verranno pubblicati, saranno certamente compensati a partire da lire 100 mila in su (a seconda del tipo di programma). Se sei fantasioso, bravo e veloce oltre che ordinato, puoi arrivare a guadagnare un bel gruzzoletto! Naturalmente il tuo nome apparirà stampato sulla rivista come valido collaboratore. Fruga nella tua fantasia e mandaci non solo arcade ma anche utility inedite, giochi di società particolari, interessanti routine in linguaggio macchina ed avventure con grafica (invia sempre anche la soluzione comando per comando in sequenza) allegando anche caratteristiche tecniche dei programmi, listati, stampate di screen etc. Se decidi di inviare un programma (indirizza a Load'n'Run, c.so Vitt. Emanuele 15, 20122 Milano) segui queste regole:

- 1) salva il programma su entrambi i lati di una cassetta;
- 2) usa 1 cassetta per ogni programma;
- 3) scrivi il tuo nome, quello del programma e quello della macchina sulla cassetta;
- 4) accludi un foglio dattiloscritto dove spiegherai a cosa serve il programma, come si usa, quali tasti usare, lo scopo. Non proteggete i programmi! Dobbiamo guardarli dentro. Non mettete il vostro indirizzo all'interno del programma (basta «by... nome e cognome»).

Causa diversità hardware tra le differenti serie di Spectrum, mandaci programmi che non usino il comando IN per la lettura della tastiera; sostituisilo con un'altra istruzione Basic (per esempio INKEY\$) o con una routine in L/M.

Rispondiamo sempre a tutti e, soprattutto, manteniamo la parola data. Coraggio dunque, fatevi vivi!



RACCOLTA DI PROGRAMMI SU CASSETTA PER IL TUO COMPUTER

SPECTRUM COLLEZIONE

N. 2 - NOV./DICEMBRE 1985 - L. 12.000

by **LOAD'N'RUN**

2

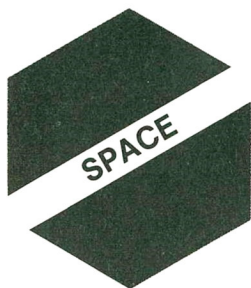
CASSETTE

GIOCHI - UTILITY - AVVENTURE

RACCOLTA
DEI PROGRAMMI
PUBBLICATI NELL'84/85

Sped. in abb. post. Gr. III/70

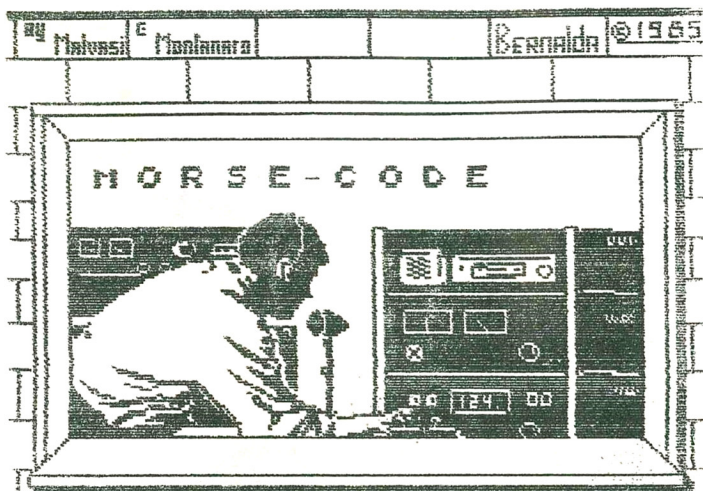
↑ **CHIEDI IN EDICOLA** ↑
ANCHE QUESTO FASCICOLO!



LUNE (48K)

Sei il leggendario cavaliere dello spazio e devi conquistare le quattro lune della tua costellazione. Per compiere la tua impresa devi atterrare su una luna, evitare e distruggere le meteore ed infine recuperare gli uomini. Cerca di recuperare più uomini possibile perché così facendo potrai avanzare nella tua impresa. Più resisterai e più faticosa sarà la tua conquista; la velocità aumenterà, i nemici della luna saranno sempre di più e tu, cavaliere dello spazio, non dovrai distrarti se vorrai essere il vincitore dell'impresa.

L = cambia il livello di gioco (da 1 a 4)
K = seleziona 1 o 2 giocatori
H = riparte il gioco
Enter = suono on/off
O = sinistra
P = destra
Q-T = incrementa la velocità sulla luna
A-G = attiva la difesa spaziale. Sulla luna, invece, diminuisce la velocità
CAPS-SPACE = fuoco
Per giocare si possono utilizzare anche i joystick tipo Kempston o Interface 2.



(48K) MORSE CODE

di G. Montanaro
e D. Malvasi

Se vuoi imparare il codice Morse o vuoi allenarti nella trasmissione diretta, non puoi fare a meno di questo programma. Esso è composto da 5 opzioni:

- 1) permette di variare la frequenza del beep; premendo il tasto «p» si possono ascoltare le variazioni.
- 2) Consente di variare la velocità di trasmissione del testo; premendo il tasto «p» si possono ascoltare le variazioni.
- 3) Il computer è pronto per la trasmissione diretta; bisogna inserire il testo da trasmettere (massimo 3999 caratteri) e terminare l'input con ENTER; infine, bisogna premere un tasto per dare il via alla trasmissione. Attenzione: durante l'inserimento del testo non è possibile cancellare l'ultimo carattere digitato!
- 4) Istruzioni.
- 5) Questa è la parte dedicata al-

l'apprendimento. Sul video sono visualizzati tutti i caratteri che possono essere trasmessi ed un cursore lampeggiante scorrerà a fianco dei caratteri al comando dei tasti 6 e 7. Premendo il tasto «p» si ascolterà la traduzione in codice Morse del carattere puntato dal cursore.

Andiamo ora a notare qualcosa nel programma (visto lo screen che bello?!).

Quando si deve spostare una grande quantità di dati da una zona di memoria ad un'altra il Basic dello Spectrum si rivela piuttosto lento, soprattutto se lo spostamento deve avvenire in un tempo assai breve. Per risolvere questo inconveniente si ricorre al linguaggio macchina dello Z80. Quest'ultimo prevede un'istruzione assai potente: LDIR. È un'istruzione appartenente al set di comandi di ge-

stione blocchi-dati. Per poter funzionare la LDIR si avvale dei tre registri HL, DE e BC. Il suo funzionamento è il seguente:

- 1) trasferisce il byte indirizzato dal registro HL nella locazione di memoria puntata dal registro DE.
- 2) Incrementa HL e DE ed infine decrementa BC.
- 3) Se il valore di BC è 0 il microprocessore esegue la successiva istruzione altrimenti ricomincia dal punto 1.

Tutte queste operazioni vengono eseguite da una sola istruzione!

Il programma caricatore di «MORSE CODE» utilizza proprio questa istruzione per trasferire in memoria il contenuto della VRAM cioè di quella parte della memoria dedicata al video, nella quale sono memorizzati i dati dei caratteri, dei disegni, dei colori ecc.

Nel listato Basic le linee 30-42 contengono i codici macchina del listato Assembler. Tali codici vengono caricati in memoria a partire dalla locazione 47537 fino all'indirizzo 47584. Alle linee 75, 85, 100 e 110 l'istruzione RANDOMIZE USR esegue il linguaggio macchina. Osservando il listato Assembler si può vedere l'uso frequente dell'istruzione LDIR. Inizialmente viene utilizzata per trasferire i dati dalla VRAM alla memoria; questo pezzo di programma viene eseguito subito dopo aver caricato uno screen. Poi l'istruzione LDIR serve a trasferire lo screen precedentemente memorizzato in VRAM, il tutto alla velocità impressionante di mezzo decimo di secondo!

```

B9BA EDB0      LDIR
B9BC C9        RET
B9BD 21D9E0    LD HL, E0D9
B9CD 110040    LD DE, 4000
B9CE 01001B    LD BC, 1B00
B9CF EDB0      LDIR
B9D0 C9        RET
B9D1 210040    LD HL, 4000
B9D2 1100C4    LD DE, C400
B9D3 01001B    LD BC, 1B00
B9D4 C9        LDIR
B9D5 2100C4    RET
                LD HL, C400

```

```

30 DATA 33,0,64,17,217,224,1,0
0,27,237,176,201
40 DATA 33,217,224,17,00,64,1,
0,27,237,176,201
41 DATA 33,00,64,17,00,196,1,0
0,27,237,176,201
40 DATA 33,00,196,17,00,64,1,0
0,27,237,176,201

```



CALCIOMANIA (48K)

di A. Balsotti



In questo periodo si sta svolgendo il campionato di calcio ed anche tu certamente vorrai tenere ordinati i risultati di ogni partita per poter poi sviluppare una possibile schedina vincente!

Il programma si divide in diversi menu:

Inizializzazione: 1) 1° giornata: il programma provvede a creare un file vuoto. 2) Altre giornate: il programma carica il file dati. 3) Restart: ha l'effetto del comando RUN.

Menù principale: si divide in: 1) Visualizzazione classifiche. 2) Aggiornamento. 3) Salvataggio. 4) Pronostico.

Visualizzazione classifiche.

Permette di vedere sullo schermo le varie classifiche: serie A, serie B o entrambe.

Ecco l'elenco delle abbreviazioni usate nella visualizzazione: Pu=punti; Gi=giocate; Vi=vinte; Nu=pareggiate; Pe=perse; Rf=reti fatte; Rs=reti subite; Mi=media inglese.

Aggiornamento dati.

In questo sottomenù è possibile modificare il file dati dopo nuove giornate di campionato.

Salvataggio dati.

Dopo aver aggiornato il file dati, è indispensabile salvarlo (si può scegliere il salvataggio su microdrive o registratore).

Pronostico schedina.

Inizialmente, se non vi è già in memoria la schedina, bisogna formulare le differenti partite del concorso del totocalcio. Successivamente si può scegliere se si vuole il pronostico del computer o semplicemente sviluppare il sistema ridotto di una propria schedina.

N.B. Se in un qualsiasi momento in cui è richiesta la pressione di un tasto si preme EDIT (Caps+1), si ottiene l'hard-copy sulla stampante.

Il programma è okay anche per microdrive! Che come è noto è una delle periferiche più importanti per lo Spectrum. Un lettore di cartucce magnetiche ad alta velocità. Le sue prestazioni si collocano tra quelle di un registratore a cassette e quelle di un disk drive.

Per poter collegare il Microdrive allo Spectrum occorre disporre anche della Interface 1 la quale oltre a interfacciare lo Spectrum al nuovo supporto di massa, dispone anche di una interfaccia seriale

```

3099>REM *** Operazioni con microdrive ***
3100 LET b$="cartuccia nel microdrive": GO SUB 3200
3110 ERASE "M";1:"Dati Toto":S
AVE #"M";1:"Dati Toto" DATA f$(
): GO SUB 3250: LET b$="": GO S
UB 3275: VERIFY #"M";1:"Dati To
to" DATA f$( ): GO SUB 3350: GO T
O 3000

```


RS-232 nonché della possibilità di collegare fino a 64 Spectrum tra loro.

Numerosi sono i nuovi comandi che vengono ad essere disponibili con l'Interface 1.

Questo programma prevede non solo la memorizzazione dei dati su cassetta ma anche su Microdrive. I comandi dedicati a questa periferica sono tutti raggruppati nella linea 3100.

Osserviamo tale linea e notiamo tre nuove istruzioni: ERASE, SAVE * e VERIFY *.

L'istruzione ERASE si presenta nella seguente forma sintattica: ERASE «m»;y;«nome file» dove y è il numero del Microdrive sul quale l'istruzione agisce; e «nome file» è il nome del blocco dati da cancellare. Quindi questo comando viene utilizzato per cancellare un programma, un blocco dati, uno screen ecc. precedentemente registrato sul Microdrive.

Le istruzioni SAVE *, VERIFY * e LOAD * vengono usate aggiungendo dopo il segno * i seguenti parametri: «m»;y;«nome file» dove «m» indica il supporto microdrive (può anche essere «n»

```

3200 LET x=12: LET y=159: LET a$
="Inserisci la "+b$: GO SUB 9990
: LET x=36: LET y=143: LET a$="D
: Remi un tasto per continuare": G
O SUB 9990: LET x=22: LET y=135:
LET a$="0 0 per tornare al menu
precedente": GO SUB 9990
3300 LET l1=0: LET l2=255: GO SU
B 9990
3400 IF i$="0" THEN GO TO 3000
3500 CLS : FLASH 1: INK 5: LET x
=79: LET y=88: LET a$="Salvatagg
io Dati": GO SUB 9990: FLASH 0
3640 RETURN
3950 CLS : LET x=52: LET y=159:
LET a$="Vuoi verificare la cor-
": GO SUB 9990: LET y=152: LET
a$="retta registrazione?(S/N)":
GO SUB 9990
4000 LET l1=0: LET l2=0: LET k1=
79: LET k2=83: LET k3=0: GO SUB
9990: IF i$="N" THEN GO TO 100
4070 RETURN
4275 CLS : LET x=70: LET y=159:
LET a$=b$: GO SUB 9990
4300 LET x=0: LET y=143: LET a$=
"Se viene segnalato un errore di
registrazione ritenta il salva
taggio con GOTO LINEA": GO SUB 9
990: LET x=40: LET y=115: LET a$

```

per network o «b» per canale seriale), y il numero del microdrive e «nome file» il nome del programma, codici o array che si vuole salvare, verificare o caricare. In questi ultimi due casi dopo «nome fi-

le» andranno rispettivamente i comandi CODE o DATA.

Molti altri comandi sono disponibili con questa interfaccia tra i quali ricordiamo: CAT, CLOSE, FORMAT, OPEN, MOVE ecc.



THE STRONG BOX (48K)

di F. Marozzo



Eccoti nei panni di uno scassinatore il cui grimaldello è il computer. Sei riuscito con il tuo micro ad inserirti sulla linea che comanda la combinazione di una cassaforte contenente documenti molto importanti. Adoperando i tasti «1», «2», «3» e «Q», «W», «E» dovrai tentare di formulare la giusta combinazione per poter accedere ai documenti. La combinazione è un numero casuale di tre cifre diverso da zero. Attenzione: se verrà premuto un tasto adiacente a quello relativo ad un numero esatto della combinazione, il computer emetterà un «click», proprio

come in una normale serratura. Sbrigati però perché il tempo passa!

Esaminiamo ora il programma.

Una delle nuove istruzioni inserite nelle ultime versioni del Basic (compresa quella dello Spectrum) è DEF FN cioè definizione di funzione.

La struttura generale di questa istruzione è:

DEF FN x ([y1],...[yn])=espressione

dove x e y1...yn indicano le variabili. Mentre x è obbligatorio definirla (è il nome della funzione!), la variabile y1...yn si può ometterla.

Quest'ultima è il parametro che viene passato alla funzione per il calcolo dell'espressione. La funzione restituisce sempre un valore che può essere numerico o alfanumerico. Vediamo subito un esempio prendendo in considerazione il listato di questo programma. Dalla linea 130 alla linea 170 sono definite tre funzioni assegnate, rispettivamente, alle variabili a, b e c.

Queste funzioni vengono richiamate durante la lettura della tastiera (linea 270) e il loro risultato consente di formare il codice ASCII dei numeri della combina-

zione visualizzati in basso sullo schermo. Osserviamo la linea 270, in k\$ vi è il tasto premuto, successivamente l'istruzione LET a=a+FN a() assegna ad a il suo valore iniziale (48 che è il codice di «0») più quello restituito dalla chiamata alla funzione a (FN a()). La DEF FN a (linea 130) è strutturata in modo tale da restituire tre valori: 0,1 e -1. Restituisce 0 quando non è stato premuto il tasto «1» o «q» o quando è stato premuto il tasto «1» ma la prima cifra della combinazione visualizzata è 9, oppure quando è stato premuto il tasto «q» ma la cifra visualizzata è 0. L'FN a() restituisce il valore 1 se il tasto premuto è «1» (k\$="1") e il numero visualizzato è minore di 9 (a<57); restituisce invece il valore -1 se il tasto digitato è «q» e il numero visualizzato è maggiore di 0 (a>48).

L'uso delle funzioni è molto importante perché consentono di risparmiare memoria e vengono usate direttamente nelle assegnazioni.

```

10>POKE 23607,176: POKE 23624,
57: RANDOMIZE USR 45300
30 LET c=200
50 RANDOMIZE USR 45250: PRINT
#0: INK 7: PAPER 1: FLASH 1):
premi c per cominciare
RANDOMIZE USR 45085: INPUT "": R
RANDOMIZE USR 45001: RANDOMIZE US
R 45238:
70 RANDOMIZE USR 44920: PAUSE
50: RANDOMIZE USR 44920
90 RANDOMIZE
110 DIM a$(3): DIM b$(3): LET a
=48: LET b=a: LET c=a
130 DEF FN a()=0+(1 AND k$="1"
AND a<57)-(1 AND k$="q" AND a>48
)
150 DEF FN b()=0+(1 AND k$="2"
AND b<57)-(1 AND k$="w" AND b>48
)
170 DEF FN c()=0+(1 AND k$="3"
AND c<57)-(1 AND k$="e" AND c>48
)
190 FOR n=1 TO 3: LET b$(n)=CHR
# (INT (RND*10)+48)
210 NEXT n
230 IF b$="000" THEN GO TO 190
250 REM
270 LET k$=INKEY$: LET a=a+FN a
(): LET b=b+FN b(): LET c=c+FN c
()
290 LET a$(1)=CHR# a: LET a$(2)
=CHR# b: LET a$(3)=CHR# c
310 IF a$(1)=b$(1) AND (k$="2"

```



MARLOWE INDAGA (48K)

di L. Martiradonna



Al ritorno da un viaggio nel regno dei Maya, i coniugi Martelli portarono come souvenir due statuette, una delle quali dotata di prodigiosi poteri. Due ladri incalliti, tentando di venire in possesso della statuetta, uccisero il marito. La moglie, presa dal panico e convinta di aver perso per sempre la statuetta, ingaggia il famoso ispettore Marlowe. Il tuo scopo è quello di scoprire dove è nascosta la statuetta. Numerosi sono i verbi ed i nomi che puoi utilizzare. Per esempio, alcuni verbi ammessi sono: prendi, lascia, entra, esci ecc. Alcuni nomi utilizzabili sono: giornale, telefono, agenda, sta-

```

3480>REM procedure find
3481 FOR n=13 TO 21
3482 PRINT AT n,0: "
3483 NEXT n
3484 RETURN
3499 REM procedure find
3500 INPUT "? "; LINE r$
3501 GO SUB 3480: LET flag=0
3502 IF r$="nord" OR r$="n" THEN
LET flag=1: IF p(x,y-1)<>0 THEN
LET y=y-1: RETURN
3503 IF r$="sud" OR r$="s" THEN
LET flag=1: IF p(x,y+1)<>0 THEN
LET y=y+1: RETURN
3504 IF r$="est" OR r$="e" THEN
LET flag=1: IF p(x+1,y)<>0 THEN
LET x=x+1: RETURN
3505 IF r$="ovest" OR r$="o" THE
N LET flag=1: IF p(x-1,y)<>0 THE
N LET x=x-1: RETURN

```


tuetta ecc. Buona fortuna... Marlowe!

C'è qui una subroutine che vogliamo farvi notare. Andiamo con ordine.

Il comando CLS dello Spectrum serve a cancellare ogni disegno e scritta presente sullo schermo e a posizionare la prossima stampa a partire dalla colonna 0 e riga 0, cioè in alto a sinistra dello schermo.

Alcune volte è necessario cancellare solo una parte dello schermo e mantenerne un'altra proprio come in questo programma. L'uso dell'istruzione CLS cancellerebbe anche quella zona dello schermo

che deve rimanere invece presente.

Questo programma deve cancellare le linee della riga 13 fino alla riga 21. Un'apposita subroutine è stata inserita nel programma per svolgere questa funzione. La subroutine risiede dalla linea 8480 alla linea 8484. Il suo funzionamento è assai semplice; un ciclo FOR-NEXT con valore minimo 13 e valore massimo 21 controlla l'istruzione PRINT AT n,0 (linea 8482) che stampa un'intera riga di spazi cancellando quanto in essa contenuto. Dato che l'istruzione PRINT opera dalla linea 13 alla linea 21 il risultato ottenuto è quello di aver cancellato questa

parte bassa dello schermo.

Vi è però un sistema più facile e soprattutto più veloce di cancellare le righe 13-21. Bisogna agire sulla variabile di sistema DF SZ (23659) la quale contiene il numero di linee riservate all'input e all'edit. Normalmente questo valore è 2 però modificandolo a 11 ed eseguendo l'istruzione INPUT" " si ottiene l'effetto di cancellare le righe 13-21.

Per eseguire questa modifica nel programma bisogna digitare la seguente linea:

8483 POKE 23659,11: INPUT" " e cancellare le linee 8481 e 8482. ■



MERGE SCREEN (16K)

di E. Dassi

Quante volte avresti voluto unire due o più screen in uno solo?! Con questa utility il tuo Spectrum potrà eseguire dei veri e propri «fotomontaggi» senza però considerare i colori. Infatti per fare il merge degli screen, questi devono essere salvati con il seguente comando:

SAVE «nome screen» CODE 16384,6144

La routine è completamente rilocabile ed occupa solo 169 bytes.

Per comodità di programmazione ti consigliamo di caricarla nel printer buffer.

Per eseguire il merge di due o più screen, carica la routine con il comando: LOAD «MERGEVIDEO» CODE n, dove n indica l'indirizzo d'inizio in RAM. Per caricare gli screen successivi dare per ogni merge il seguente comando: RANDOMIZE USR n, dove n è l'indirizzo di partenza della routine in RAM.

Questo programma utilizza esclusivamente un caricatore (LOAD) in I/m per ottenere il merge di più screen. La caratteri-

stica principale di questa routine è quella di essere rilocabile, cioè è possibile caricarla a qualsiasi indirizzo ed eseguirla normalmente con il comando RANDOMIZE USR n, dove n indica l'indirizzo d'inizio.

Il programma sfrutta per il caricamento lo stesso algoritmo residente in ROM, con la sola differenza che il load del nuovo dato non cancella quello vecchio bensì ne esegue un «merge» tramite l'istruzione logica assembler OR. Questa si trova nel listato all'indirizzo C3C0H: OR (IX+00). Il registro indice IX punta alla locazione dove inserire il dato caricato e l'accumulatore contiene il dato caricato da nastro. Dopo aver

eseguito l'OR, nell'accumulatore vi sarà il «merge» del byte che dovrà essere copiato nel registro L e successivamente memorizzato all'indirizzo (IX+00). Numerose sono le possibili modifiche a questa breve ma efficace routine. Per esempio anziché eseguire l'istruzione OR (IX+00) si può inserire al suo posto il comando AND (IX+00) ottenendo così non più un merge tra gli screen ma un «AND».

Allo stesso modo si può sostituire l'OR con l'istruzione XOR (IX+00) oppure se desiderate caricare uno screen in inverse utilizzare l'istruzione CPL al posto di OR (IX+00). ■



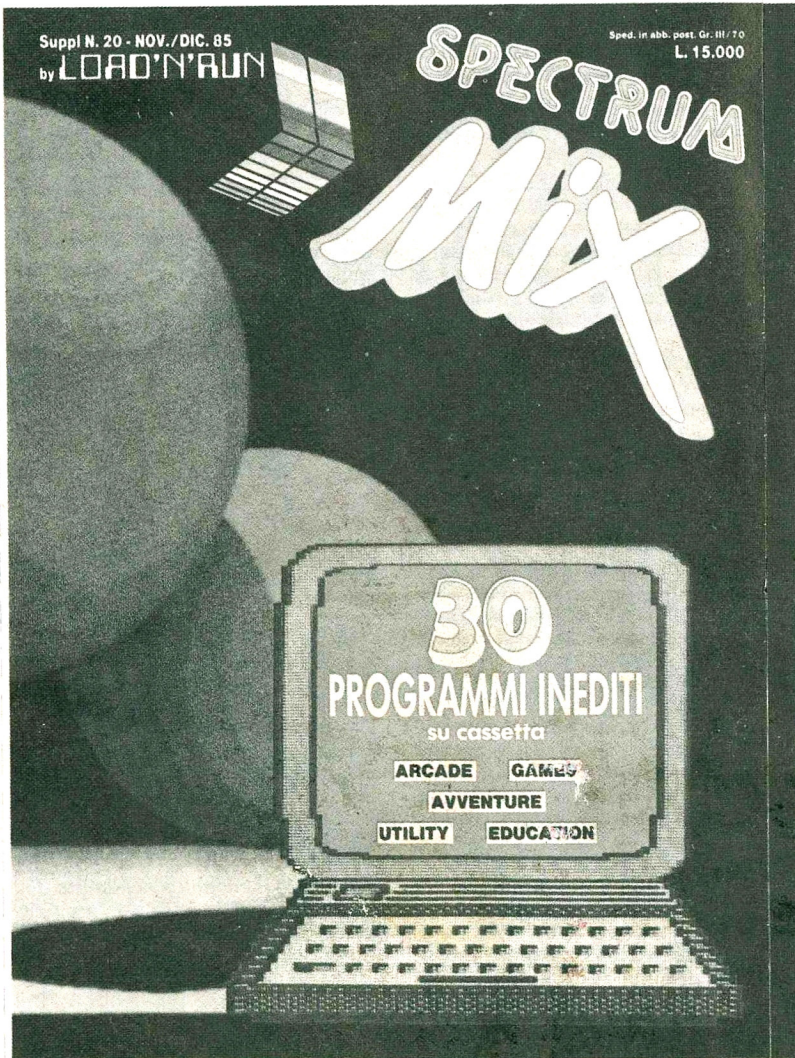
```

00000 DD210C58 LD IX,5BCC
00004 111100 LD DE,0011
00007 AF XOR A
00008 37 SCF
00009 DD5006 CALL 0556
0000D DD210040 LD IX,4000
0000E 11001040 LD DE,1000
0000F 3E7F LD A,7F
00010 37 SCF
00014 14 INC D
00017 00 EX AF,AF

```


by **LOAD'N'RUN**
SPECIALE

30
PROGRAMMI
INEDITI!



in edicola!

LA PIÙ BELLA
RIVISTA
DI ELETTRONICA
& COMPUTER

★ ★ ★

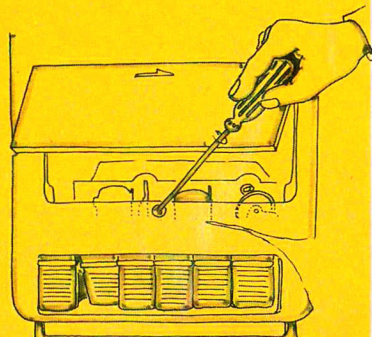
progetti
programmi
hardware
software



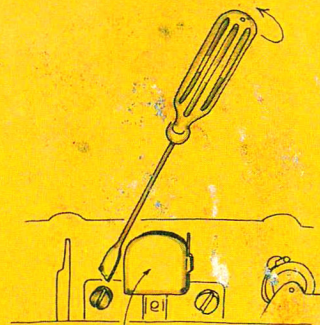
Electronica 2000 MISTER KIT

ogni mese
in tutte
le edicole

LE NOSTRE CASSETTE SONO PERFETTE



Se la cassetta non funzionasse al primo colpo, prova a regolare (vedi disegno!) la vite della testina di lettura del registratore fino ad ottenere un segnale perfetto nell'altoparlante!



TESTINA DI
LETTURA

- 1) **VOLUME.** Il volume del registratore deve essere sufficientemente alto, in modo da sentire chiaramente dal buzzer del computer il programma in fase di caricamento.
- 2) **TONO.** Se il registratore ha una sola manopola per la regolazione del tono, essa va spostata tutta verso gli alti (high). Se invece è possibile una equalizzazione, bisogna preferire le frequenze medio-alte.
- 3) **CANALI.** La cassetta è registrata in mono, cioè su entrambi i canali. In caso di difficoltà è opportuno provare a caricare un solo canale.
- 4) **AZIMUT.** Per la regolazione dell'azimut, occorre inserire una cassetta nel registratore, premere PLAY e regolare la vite della testina centrale fino ad ottenere dall'altoparlante il massimo segnale.